BEx Reporting on 12 Months Rolling Periods using Posting Period (0FISCPER3)

Applies to:

SAP BW 3.5, BI 7.0. For more information, visit the Business Intelligence homepage.

Summary

This article covers the basic workaround for frequently used reporting scenario that is “Rolling Periods”. It becomes more difficult when you have used posting periods (OFISCPER3) in your info provider structure. As it does not provide automatic functionality of rolling periods as fiscal year/period (0FISCPER) provides. So if you follow the below steps, you will also be able to do reporting in scenario where posting periods are used.

Author: Purvang Zinzuwadia

Company: Patni Computer Systems Ltd

Created on: 20 November, 2008

Author Bio

Purvang Zinzuwadia is working with Patni for more than 2 years. He is currently working on SAP BI 7.0. He has also worked on different areas in BI like modeling, reporting, production support, ETL process including some other areas like information broadcasting.

Table of Contents

Introduction

Here is the description of scenario. Your organization needs to do reporting on last one year data but on rolling period basis. That means user will input any month/year combination and starting from that month/year, you will have to report on previous 12 months. And you have come to know that it is not straight forward as instead of fiscal year/period (0FISCPER), your info provider contains fiscal year (0FISCYEAR) and posting periods (0FISCPER3). So below are the steps to follow and successfully achieve the desired result.

Frontend Setting

First of all you need to design your query in such a way that it can provide the right platform to achieve the desired output. Basic requirement for report is that user needs to input at least one month/year combination. For this you will require two user defined variables. Create them as below.

Create variables for posting period (0FISCPER3)

Create a variable named ZPER_P0 for posting period to store the initial value input by user as per below specifications using create variable wizard.

- Variable Type: Characteristic Value
- Processing Type: User Entry/Default Value
- Variable Entry: Mandatory
- Ready for Input selected

Now, create another 11 variables named ZPER_P1 to ZPER_P11 for posting period to store the rest of eleven values calculated automatically as per below specifications using create variable wizard.

- Variable Type: Characteristic Value
- Processing Type: Customer Exit
- Variable Entry: Mandatory
- Ready for Input **NOT** selected

Create variables for fiscal year (0FISCYEAR)

Create a variable named ZFYR_Y0 for fiscal year to store the initial value input by user as per below specifications using create variable wizard.

- Variable Type: Characteristic Value
- Processing Type: User Entry/Default Value
- Variable Entry: Mandatory
- Ready for Input selected

Now, create another 11 variables named ZFYR_Y1 to ZFYR_Y11 for fiscal year to store the rest of eleven values calculated automatically as per below specifications using create variable wizard.

- Variable Type: Characteristic Value
- Processing Type: Customer Exit
- Variable Entry: Mandatory
- Ready for Input **NOT** selected

This finishes our front-end setting. Now it is time to fill those customer exit variables. Follow the below section of backend setting.

Backend Setting

Now we have created the necessary variables to do reporting on rolling periods. We have created one user entry variable for both posting period and fiscal year and eleven customer exit variable for both posting period and fiscal year. To fill those customer exit variable, now we need to write the code in RSR00001 exit (Customer Exit Global Variables in Reporting). To accomplish this task basic knowledge of ABAP is required. Follow the below steps to fill the values for variables.

- Create a project in CMOD (if you do not have any)
- Assign the enhancement RSR00001 to the project
- Save and activate the project
- Go to SMOD transaction
- Select RSR00001 as enhancement
- Select components and click Display
- Double click on EXIT_SAPLRRS0_001 function module
- Double click on ZXRSRU01 include

A code editor screen will appear; click on Change button icon to enable the code editor for input.

Logic for filling Posting Period Values

Now we want to fill the rest of the eleven posting period variables ZPER_P1 to ZPER_P11 in rolling back fashion. That means if user inputs value of posting period variable at selection screen as ‘003’ (March), than ZPER_P1 should have ‘002’ (February), ZPER_P2 should have ‘001’ (January), ZPER_P3 should have ‘012’ (December, of previous year) and so on.

To accomplish the above task logic is given below.

Step 1: Read the user input value at selection screen for first posting period variable i.e. ZPER_P0

Step 2: Subtract the index number of current processing variable from the user input value (Refer to code)

Step 3: If the subtraction is less than 1 than add 12 to the result

Step 4: Assign the result as value to currently being processed customer exit variable for posting period

Sample code for ZPER_P1 posting period variable

```
read table i_t_var_range into loc_var_range with key vnam = 'ZPER_P0'. “(Reference to
step 1)
  if sy-subrc = 0.
    per = loc_var_range-low.
  endif.
  nper = per+58(2).
  nper = nper - 1. “(reference to step 2, here 1 is subtracted because ZPER_P1)
  if nper < 1.
    nper = nper + 12. “(Reference to step 3)
  endif.
  per = nper.
  lastper = per.
  l_s_range-sign = 'I'.
  l_s_range-opt  = 'EQ'.
  l_s_range-low = lastper+57(3). “(Reference to step 4)

  append l_s_range to e_t_range.
```

Logic for filling Fiscal Year Values

At this point we have filled correctly the posting period values but what about the corresponding year values. We also need to fill these fiscal year values correctly. For example, if month is getting changed from January to December than year should also change from 2008 to 2007. Now we will see in the following section how we populate ZFYR_Y1 to ZFYR_Y11 fiscal year variables.

To accomplish the above task, logic is given below.

Step 1: Read the user input value at selection screen for first fiscal year variable i.e. ZFYR_Y0

Step 2: Read the user input value at selection screen for first posting period variable i.e. ZPER_P0

Step 3: Subtract the index number of current processing variable from the user input value of posting period (Refer to code)

Step 4: If the result of posting period subtraction is less than 1 than subtract the fiscal year value by 1 (fiscal year is subtracted always by 1 for all variables created on fiscal year).

Step 5: Assign the new fiscal year value to currently being processed customer exit variable for fiscal year

Sample code for ZFYR_Y1 posting period variable

```
read table i_t_var_range into loc_var_range with key vnam = 'ZFYR_Y0'. “(Reference to
step 1)
if sy-subrc = 0.
  l_fiscyear = loc_var_range-low.
  read table i_t_var_range into l_posper_rng with key vnam = 'ZPER_P0'. “(Reference
to step 2)
  if sy-subrc = 0.
    l_posper_rng-low = l_posper_rng-low - 1. “(Reference to step 3)
    if l_posper_rng-low < 1. “(Reference to step 4)
      l_fiscyear = l_fiscyear - 1. “(Reference to step 5)
    endif.
  endif.
endif.
l_s_range-sign = 'I'.
l_s_range-opt  = 'EQ'.
l_s_range-low = l_fiscyear.
append l_s_range to e_t_range.
```

Create similar codes for rest of the variables for posting period and fiscal year. Do not forget to keep in mind the number of posting period or fiscal year variable you are processing e.g. index number of variable ZPER_P3 is 3.

Save and activate the code and project and now you are ready to report on rolling periods.

Enhancements

- You can also use function module to omit the repeated block of logic e.g. subtracting the value from posting period variable
- You can also use text variables to display the descriptive titles for month names like February, January etc. instead of pure numbers like 2, 1 etc. in the report output.

Related Content

- http://help.sap.com/
- https://www.sdn.sap.com/irj/sdn

For more information, visit the Business Intelligence homepage.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.